Functional Differences Between Masks 5H80G and 0J17D of the DSP56309

This document describes the differences between masks of the DSP56309: the 0J17D and the mask immediately preceding it, 5H80G. The 0J17D mask of the DSP56309 uses the communications design rules (CDR2) process. The 0J17D mask set also has new I/O and a new PLL, with the requisite change in the PLL capacitor equation (PCAP).

CONTENTS

1 PLL Input Capacitor

The process change results in a changed requirement for computing the size of C_{PCAP}, the capacitor used with the PCAP input. **Table 1** lists the new formulas for computing the value of this input capacitor for the DSP.

2 ESD Protection

The 0J17D mask set incorporates improved I/O structures to ensure Electro Static Discharge (ESD) protection to Freescale's MC qualification levels of 2000 volts Human Body Model (HBM) and 200 volts Machine Model (MM).

3 Differences Overview

The primary functional differences between the masks are due to inherent differences between the two design technologies. **Table 1** compares the mask sets.

Table 1. Functional Comparison of 5H80G and 0J17D

Feature	5H80G (80 MHz)			0J17D (100 MHz)		
	Recommended	Min	Max	Recommended	Min	Max
Technology	0.5 micron	—	—	Sub 0.4 micron	—	—
PLL input capacitor (C_{PCAP})	Uses the following rules: For MF ≤ 4: C_{PCAP} = [(500 × MF) – 150] pF For MF > 4: C_{PCAP} = (690 × MF) pF	(MF × 425) – 125 MF × 520	(MF × 590) – 175 MF × 920	Use the following rules: For MF ≤ 4: C_{PCAP} = [(680 × MF) – 120] pF For MF > 4: C_{PCAP} = (1100 × MF) pF	(MF × 580) – 100 MF × 830	(MF × 780) – 140 MF × 1470

Key differences between the masks are as follows:

- Target operating frequency for the 0J17D mask set is 100 MHz full voltage range (3.0–3.6 V). The preceding 5H80G mask set was 80 MHz full voltage range (3.0–3.6 V) and 100 MHz with de-rated operating voltage (3.125–3.6 V).
- On the 0J17D mask set, the peripheral I/O are 5 V tolerant without keepers. The preceding 5H80G mask set did not have 5 V tolerant I/O and had keepers.
- The J17D mask set has split voltage supply between core and I/O supplies.
- Pull-up was removed from the TCK pin in order to make the DSP56309 consistent with the DSP563xx family. The preceding 5H80G mask set had a pull-up on the TCK pin

4 Identification Registers

Values in the Device Identification register (IDR) and JTAG Identification (ID) register are changed to reflect the new 0J17D mask set, as shown in **Figure 1** and **Figure 2**. The IDR is a 24-bit, read-only factory-programmed register that identifies DSP56300 family members. It specifies the derivative number and revision number of the device. This information is used in testing or by software. **Figure 1** shows the contents of the IDR for mask set 0J17D. Revision numbers are assigned as follows: $0 is revision 0, $1 is revision A, and so on.

23 16	15 12	11 0
Reserved	**Revision Number**	**Derivative Number**
0000 0000	0001	0011 0000 1001

Figure 1. Identification Register Configuration (Revision 3)

The JTAG ID register is a 32-bit read-only factory-programmed register that distinguishes the component on a board according to the IEEE 1149.1 standard. **Figure 2** shows the JTAG ID register configuration for mask set 0J17D. Version information corresponds to the revision number ($0 for revision 0, $1 for revision A, and so forth).

31 28	27 22	21 12	11 1	0
Version Information	**Customer Part Number**	**Sequence Number**	**Manufacturer Identity**	**1**
0001	000110	0000001001	000 0000 1110	1

Figure 2. JTAG Identification Register Configuration (Revision 3)

5 Errata Removal

The 0J17D mask set removes all known functional errata on the 5H80G mask set. This section shows the functional errata that the 0J17D mask set removes. For further documentation updates, consult the errata for each device. The specific differences in errata items are listed in the following tables. A detailed description of each errata item is available in the chip errata documents on the web site listed on the back cover of this document.

The silicon errata present on the 5H80G mask but not on the 0J17D masks are listed as follows:

Errata Number	Errata Description	Status
ES33	Description (added 3/3/1997): When using the JTAG instructions SAMPLE/PRELOAD, EXTEST, and CLAMP, erroneous data may be driven out on the parallel pins and TDO. Data cannot be shifted through the Boundary Scan Register (BSR) using the SAMPLE/PRELOAD instruction. Because the BSR must be preloaded using the SAMPLE/PRELOAD instruction, the EXTEST and CLAMP instructions cannot be used for testing the board connections. Workaround: None available.	**Fixed on 0J17D**
ES42	Description (added 2/27/1998): When a DMA channel is in Line mode (that is, DTM = 010) with address modes defined by D3D = 0 and DMA = 10010x (that is, DCO in A mode), and the DCO value is greater than $FFF, then the DMA does not function properly. This address mode implies “no update” at the destination and “no update” or “post increment by 1” mode at the source. Workaround: Use Block Transfer mode (that is, DTM = 000). For the DCO and DAM settings described in this erratum, the Line Transfer mode of DMA is identical to its Block Transfer mode, so this combination is redundant. In fact, a block containing only one line is still a block.	**Fixed on 0J17D**

Errata Number	Errata Description	Status
ES53	Description (added 9/25/1997): Using the JTAG instruction code 1111 ($F) or 1101 ($D) for the BYPASS instruction may cause the chip to enter Debug mode (which then correctly sets the Status bits (OS[1:0]) in the OnCE Status and Control Register (OSCR[7:6]) and asserts the $\overline{DE}$ output to acknowledge Debug mode status). Workaround: Use one of the following alternatives: • If possible, do not use instruction code 1111 ($F) or 1101 ($D) for the BYPASS instruction. Use one of the other defined BYPASS instruction codes (that is, any code from 1000–1100 ($8–$C) or 1110 ($E)). • If you must use instruction code 1111 ($F) or 1101 ($D), use the following procedure: — While the $F or $D instruction code is in the Instruction Register, ensure that the JTAG Test Access Port (TAP) state machine does not pass through the JTAG Test-Logic-Reset state while accessing any JTAG registers (that is, Instruction Register, Boundary Scan Register, or ID Register). — Before using any other JTAG instruction, load one of the other BYPASS instruction codes (that is, any code from 1000–1100 ($8–$C) or 1110 ($E)) into the instruction register. Then, any other JTAG instruction may be used.	**Fixed on 0J17D**
ES54	Description (added 2/27/1998): When a DMA channel is configured to use its DMA Control Register (DCR) in the following way: • Line Transfer mode is selected (that is, DTM[2:0] = 010). • Non-three-dimensional Address mode is selected (that is, D3D = 0). • Destination Address Offset Register DOR1 or DOR2 is selected (that is, DAM[5:3] = 001 or 011). • No Source Address Offset is selected (that is, DAM[2:0] = 100 or 101). then DMA transfers do not occur as intended. Workaround: Select Destination Address Offset Register DOR0 or DOR2 by setting DAM[5:3] = 000 or 010.	**Fixed on 0J17D**

Errata Number	Errata Description	Status
ES77	Description (added 2/27/98): The PCAP pin of the H80G is marginal to Freescale's XC qualification requirement of 1K Volt Human Body Model ElectroStatic Discharge (ESD) and 100V Machine Model ESD. Workaround: Minimize PCAP exposure to ElectroStatic Discharge.	**Fixed on 0J17D**
ES78	Description (added 2/27/98): When the SCI is configured in Synchronous mode, if the internal clock and all SCI pins are enabled simultaneously, then an extra pulse of 1 DSP clock-length is provided on the SCLK pin. Workaround: Do both these steps in this order: a. Enable an SCI pin other than SCLK. b. In the next instruction, enable the remaining SCI pins, including SCLK.	**Fixed on 0J17D**
ES79	Description (added 2/27/1998): When activity passes from one DMA channel to another and the DMA interface accesses external memory, requiring one or more wait states, the DACT and DCH status bits in the DMA Status Register (DSTR) may indicate improper activity status for DMA Channel 0 (that is, DACT = 1 and DCH[2:0] = 000). Workaround: none	**Fixed on 0J17D**

Errata Number	Errata Description	Status

ES84	Description (added 5/13/98):
	
	When software disables a DMA channel (by clearing the DE bit of the DCR) , the DTD status bit of the channel may not be set if any of the following events occur:
	
	a. Software disables the DMA channel just before a conditional transfer stall (Described by App B-3.5.1,UM).
	
	b. Software disables the DMA channel at the end of the block transfer (that is after the counter is loaded with its initial value and transfer of the last word of the block is completed).
	
	As a result, the Transfer Done interrupt might not be generated.
	
	Workaround:
	Avoid using the instruction sequence causing the conditional transfer stall (See DSP56300 UM, App B-3.5.1 for description) in fast interrupt service routines. Every time the DMA channel needs to be disabled by software, the following sequence must be used :
	
	`bclr #DIE,x:M_DCR ; not needed if DIE is cleared`
	`bclr #DE,x:M_DCR`
	`; instead of two instructions above, one 'movep' instruction`
	`; may be used`
	`; to clear DIE and DE bits`
	`movep #DCR_Dummy_Value,x:M_DCR`
	`bclr #DE,x:M_DCR`
	`nop`
	`nop`
	
	Here, the `DCR_Dummy_value` is any value of the DCR register that complies with the following requirements:
	
	• DE is set;
	• DIE is set if Transfer Done interrupt request should be generated and cleared otherwise;
	• DRS[4:0] bits must encode a reserved DMA request source (see the following list of reserved DRS values);
	
	List of reserved DRS[4:0] values (per device):
	
	• DSP56302, DSP56309, DSP56303, DSP56306, DSP56362 — `10101-11111`
	• DSP56305 — `11011`
	• DSP56301 — `10011-11011`
	• DSP56307 — `10111-11111`
		Fixed on 0J17D

Errata Number	Errata Description	Status
ES85	Description (added 5/3/98): If both the DMA channel and the core simultaneously access the same 1/4K page of internal memory (X, Y, or program), an improper DMA channel operation may occur. Workaround: Avoid simultaneous DMA and core accesses to the same 1/4K page of internal memory.	**Fixed on 0J17D**
ES89	Description (added 6/25/98): If the SCI Receiver is programmed to work with a different serial clock than the SCI Transmitter so that either the Receiver or Transmitter is using the external serial clock and the other is using the internally-generated serial clock—RCM and TCM in the SCCR are programmed differently)—then the internal serial clock generator will not operate and the SCI portion (Receiver or Transmitter) clocked by the internal clock will be stuck. Workaround: Do not use SCI with the two SCI portions (Receiver and Transmitter) clocked by different serial clocks; use either both externally or both internally clocked. Or: When using both portions of the SCI (Receiver & Transmitter), do not program different values on RCM and TCM in the SCCR.	**Fixed on 0J17D**

Errata Number	Errata Description	Status
ES90	Description (added 6/25/98)/Modified 4/19/99: A deadlock occurs during DMA transfers if all the following conditions exist: 1. DMA transfers data between internal memory and external memory through port A. 2. DMA and the core access the same internal 0.25K memory module. 3. One of the following occurs: a. The bus arbitration system is active, that is, $\overline{BG}$ is changing, not tied to ground. b. Packing mode (bit 7 in the AAR[3 - 0] registers) is active for DMA transfers on Port A. Workaround: One of the following, but workarounds 2, and 3 are valid ONLY to section 3 a of the errata - that is, not valid if packing mode is used, and workaround 4 is valid only to section 3 b of the errata - that is, not valid if bus arbitration is active. 1. Use intermediate internal memory on which there is no contention with the core. 2. Tie $\overline{BG}$ to ground, or have an external arbiter that asserts $\overline{BG}$ even if BR is not asserted. 3. Set the BCR[BRH] bit, whenever BR must be active. 4. Avoid using packing mode.	**Fixed on 0J17D**

Errata Number	Errata Description	Status

ES91	Description (added 7/22/98):
	
	If the Core reads data from the HRX while instructions are fetched from the memory Expansion Port (Port A) using 2 or more wait states, data may be lost.
	
	Workaround :
	
	There are three possible workarounds:
	
	1) The host should guarantee that there is no more than one word in the TXH:TXM:TXL-HRX data path at any time. This can be achieved if the host writes a word to the HI08 only when the TRDY flag is set (that is, the data path is empty).
	
	2) Use a service routine running from fast (that is, one wait state) external memory or internal memory to read the HRX read code; ensure that code that is fetched from slow (that is, more than 1 wait state) external memory is located at least 4 instructions after the HRX register is read. For example:
	
	`READ_HRX_DATA`
	`NOP`
	`NOP`
	`NOP`
	`NOP`
	
	Note:
	a) Interrupt requests that fetch instructions from slow external memory should be masked during this service routine. Nonmaskable interrupt (NMI) request routines must not be in external memory.
	
	b) If running from fast external memory and if a DMA channel accessing external memory is used, then the DMA may cause extra wait states to the core. Thus, the DMA should have a lower priority than the core so that the core can access the external memory with no more than 1 wait state.
	
	3) Read the HRX using one of the channels of the on-chip DMA controller.
		Fixed on 0J17D

Errata Number	Errata Description	Status

ES94	Description (added 8/10/98):
	
	Enabling any DMA channel by software for transferring a block of data (TM=011 in the channel control register) might not work properly.
	
	Workaround:
	
	Triggering of a channel for block transfer by software can be replaced by triggering of the DMA channel for block transfer by a peripheral (e.g. Timer, SCI, and so forth) that is not used while the block of data should be transferred by DMA. This can be done as follows:
	
	1. Set the DSR, DDR and DCO registers of the DMA channel according to the application case.
	
	2. Transfer mode of the DMA channel (in the DCR register) should be set to TM = 000 or TM = 100 (See Section 8.1.5.3, 563xx UM).
	
	3. DMA Request Source of the DMA channel should be set according to the chosen peripheral, which should trigger the DMA channel (see Section 8.1.5.6 56300 UM and "DMA Request Sources" Table in the CORE CONFIGURATION item of the 563xx UM).
	
	4. All others fields of the DCR register, except the DE bit, should be set according to the application case.
	
	5. Configure the peripheral to assert its DMA request line;
	
	6. Set DE bit of the DCR register.
	
	Example 1:
	
	Assuming that the SCI is not used while the block of #DCO3 words is transferred by DMA channel 3, the SCI Transmit Data (TDRE = 1, DRS[4:0] = 01111) trigger can be used instead of a software trigger for channel 3.
	
	1. Initialize DMA channel registers
	
	`movep #DSR3,x:M_DSR3`
	`movep #DDR3,x:M_DDR3`
	`movep #DCO3,x:M_DCO3`
	`bset  #0,x:M_PCRE`
	
	Now when the DMA channel is enabled, a transfer of the block begins.
		Fixed on 0J17D

Errata Number	Errata Description	Status

ES94 cont.	2. Enable DMA channel
	`movep #$867a40,x:M_DCR3 ;; enable DMA channel 3, ;; block transfer mode, ;; DRS[4:0] = 01111`
	Example 2 :
	Assuming that Timer 0 is not used while a block of #DCO3 words is to be transferred by DMA channel 3, the Timer0 (TCF0 = 1, DRS[4:0] = 10000) trigger can be used instead of a software trigger for channel 3.
	1. Initialize DMA channel registers
	`movep #DSR3,x:M_DSR3`
	`movep #DDR3,x:M_DDR3`
	`movep #DCO3,x:M_DCO3`
	2. Initialize Timer 0
	`movep #$0,x:M_TCSR0 ;;no prescaling, inv=0, mode 0, ;; no interrupt, reload is disabled`
	`movep #$0,x:M_TLR0 ;;initialize load reg.`
	`movep #$0,x:M_TCPR0 ;;initialize compare reg.`
	3. Generate DMA channel trigger
	`- option 1`
	`movep #$A48254,x:M_DCR3 ;; enable DMA channel 3, block ;; transfer mode, DE isn't ;; disabled at end of ;; transfer, triggered by ;; Timer0 (TCF0=1)`
	`bset #M_TE,x:M_TCSR0 ;; enable timer`
	`nop`
	`nop`
	`bclr #M_TE,x:M_TCSR0 ;; disable timer`
		Fixed on 0J17D

Errata Number	Errata Description	Status

ES94 cont.	- option 2
	
	`movep #$848254,x:M_DCR3 ;; enable DMA channel 3, block`
	`;; transfer mode, DE is`
	`;; disabled at the end of`
	`;; the transfer, triggered by`
	`;; Timer0`
	`;; (TCF0=1)`
	
	`bset #M_TE,x:M_TCSR0 ;; enable timer`
	`nop`
	`nop`
	`bclr #M_TE,x:M_TCSR0 ;; disable timer`
	
	- option 3
	
	`bset #M_TE,x:M_TCSR0 ;; enable timer`
	`movep #$848254,x:M_DCR3 ;; enable DMA channel 3, block transfer`
	`;; mode, DE is disabled at the end of`
	`;; the transfer, triggered by Timer0`
	`;; (TCF0=1)`
	`bclr #M_TE,x:M_TCSR0 ;; disable timer`
	
	Following are the differences between these three options:
	
	option 1 : The DMA channel should be enabled only for the first block transfer. An additional block transfers can be triggered by the following sequence:
	
	`bset #M_TE,x:M_TCSR0 ;; enable timer`
	`nop`
	`nop`
	`bclr #M_TE,x:M_TCSR0 ;; disable timer`
	
	Note:
	Execution of this sequence can be interrupted because the DMA channel is triggered after the TE bit of the TCSR0 register is set. The TE bit must be cleared no later than 33554430 DSP clock cycles after it is set. Otherwise, an undesirable trigger for the DMA channel is generated. The DMA channel must be disabled every time when Timer 0 is used for another purpose.
	
	option 2 : The whole sequence must be used every time when a block transfer is to be triggered.
		Fixed on 0J17D

Errata Number	Errata Description	Status
ES94 cont.	**Note:** Execution of this sequence may be interrupted because the DMA channel is triggered after the TE bit of the TCSR0 register is set. The TE bit must be cleared no later than 33554430 DSP clock cycles after is is set. Otherwise, an undesirable trigger for the DMA channel is generated. option 3: The whole sequence must be used every time a block transfer is to be triggered. **Note:** Execution of the first two instructions of the sequence must be uninterruptable. The TE bit of the TCSR0 register must be cleared no later than 33554430 DSP clock cycles after it is set. Otherwise, an undesirable trigger for the DMA channel is generated. Another peripheral can be used for this purpose, but taking into consideration its specific features.	**Fixed on 0J17D**
ES95	Description (added 8/15/98): If more than a single DMA channel is enabled while the DSP stays in the WAIT processing state, and triggering one of the DMA channels causes an exit from the WAIT state (See A-6.115, UM), triggering another DMA channel might cause improper DMA operation. Workaround: Assure that only a single DMA channel can be triggered during DSP WAIT state. If the application cannot guarantee this, other DMA channels should be disabled before the WAIT processing state is entered and then reenabled after WAIT state is exited.	**Fixed on 0J17D**
ES100	Description (added 10/22/98): The VDD requirements for 100 MHz operation are VDD min = 3.15 volts and VDD max = 3.6 volts. Workaround: Ensure that the VDD supply value is within the specified voltage range requirements.	**Fixed on 0J17D**

Errata Number	Errata Description	Status

ES101	Description (added 10/26/98):
	
	If the reset mode is expanded mode (for example, mode 0 or mode 8 on the DSP5630x), A MOVE (not a PROGRAM FETCH) from internal P memory to any destination may not work properly.
	
	Workaround: After each reset ($\overline{\text{RESET}}$) negation and before the first move from internal program memory, execute the following sequence:
	
	`BSET  #M_CE,sr`
	`NOP`
	`NOP`
	`NOP`
	`BCLR  #M_CE,sr`
		Fixed on 0J17D

ES104	Description (added 11/20/98):
	
	An improper operation may occur when a DMA channel uses the following transfer modes:
	
	• DTM(2:0) = 100
	
	• DTM(2:0) = 101
	
	where the DE bit is not automatically cleared at the end of block and the DMA channel is disabled by software (DE bit is cleared) while it is triggered for a new transfer.
	
	Workaround: The DMA channel should be disabled only when it cannot be triggered by a new transfer. Use one of the following alternatives:
	
	1. The system configuration must guarantee that no DMA trigger can occur while the DE bit is cleared.
	
	2.The following sequence disables the DMA channel:
	
	`a/  Wait until the DTD bit is cleared`
	`b/  Clear the DE bit`
	`c/  Wait until the DTD bit is set`
		Fixed on 0J17D

ES108	Description (added 12/12/98): The Timer's TIOx pin does not function properly in WatchDog mode while the $\overline{\text{RESET}}$ pin is asserted. The TIOx pin is tri-stated immediately after $\overline{\text{RESET}}$ assertion; it should be driven for 2.5 internal clock cycles according to the specification (section 9.4.4.1 and 9.4.4.2 of the *DSP56303 User's Manual*). Workaround: Provide external logic in order to extend the TIOx pin assertion.	**Fixed on 0J17D**

Errata Number	Errata Description	Status
ES114	Description (added 4/19/99, revised 4/30/99): A DMA channel may operate improperly when the address mode of this channel is defined as three-dimensional (D3D=1) and DAM[5:0] = 1xx 1 10 or DAM[5:0] = 01xx 10 (that is, triple counter mode is E). Workaround: Use the triple counter modes C(DAM[1:0]=00) or D(DAM[1:0]=01) instead of the E(DAM[1:0]=10) mode.	**Fixed on 0J17D**

ES115	Description (added 4/19/99):
	
	When a DMA channel (called channel A) is disabled by software clearing the channel's DCR[DE] bit, the DTD bit may not get set, and the DMA end of the block interrupt may not happen if one of the following occurs:
	
	1. There is another channel (channel B) executing EXTERNAL accesses, and the DE bit of channel A is being cleared by software at the end of the channel B word transfer - if channel B is in Word transfer mode, or at the end of the channel B line transfer - if channel B is in Line Transfer mode, or at the end of the channel B block transfer - if channel B is in Block transfer mode.
	
	2. This channel (A) is executing EXTERNAL accesses, and the DE bit of this channel (A) is being cleared by software at the end of the channel B word transfer - if channel B is in Word transfer mode, or at the end of the channel B line transfer - if channel B is in Line transfer mode.
	
	Workaround:
	
	Avoid executing a DMA external access when any DMA channel should be disabled. This can be done as follows. Every time the DMA channel needs to be disabled by software, the following sequence must be used:
	
	`;; initialize an unused DMA channel "C"`
	`movep #DSR_swflag, x:M_DSRC ;; here DSR_swflag is an`
	`;; unused X, Y or P`
	`memory`
	`;; location, should`
	`;; be initialized to`
	`;; $800000`
	`;; M_DSRC - address of`
	`the`
	`;; channel C DSR`
	`register.`
		Fixed on 0J17D

Errata Number	Errata Description	Status
ES115 cont.	movep #DDR_swflag, x:M_DDRC ;; DDR_swflag is an unused ;; X, Y or P memory ;; location, should be ;; initialized to $000000 ;; M_DDRC - ;; address of the channel C ;; DDR register . movep #TR_LENGTH, x:M_DCOC ;; see below the definition ;; of the TR_LENGTH value, ;; M_DCOC - address ;; of the channel C DCO register .movep #1f0240, x:M_DCRC ;; M_DCRB - address of the ;; channel C DCR register. ;; Set transfer mode - ;; block transfer, ;; triggered by ;; software highest ;; priority, continuous ;; mode on no-update ;; source and destination ;; address mode X memory ;; location for source ;; and destination (can be ;; chosen by ;; user accordingly to ;; DSR_swflag/DDR_swflag)	**Fixed on 0J17D**

Errata Number	Errata Description	Status
ES115 cont.	;; disable DMA channel "A" ori #3, mr ;; mask all interrupts bset #23, x:M_DCRC ;; enable DMA channel C bclr #23,x:DDR_swflag,* ;; wait until DMA channel C ;; begin transfer bclr #23, x:M_DCRA ;; disable DMA channel A nop nop jclr #M_DTDA, x:M_DSTR,* ;; polling DTD bit of the DMA ;; channel A, The TR_LENGTH value can be defined as the maximum length of the external DMA transfer——from the length of the read DMA cycle and from the length of the write DMA cycle. The length of the external read/write DMA cycle can be defined as the length of the PORTA external access. The length of the internal read/write DMA cycle can be defined in the errata case as 2 DSP clock cycles. The TR_LENGTH can be found as sum of the lengths of the DMA read and DMA write cycles.	**Fixed on 0J17D**

6 Documentation Errata

Following are errata items that have been designated as documentation errata that are specification changes for the entire DSP56300 family, including the DSP56309. Therefore, when the DSP56309 user's manual and datasheet are updated in the first half of 1999, all of these items will be documented in the new revision.

Errata No.	Description	Status

ED1	1. Description (revised 11/9/98):
	
	XY memory data move does not work properly if the X-memory move destination is internal I/O and the Y-memory move source is a register used as destination in the previous adjacent move from non Y-memory OR the Y-memory move destination is a register used as source in the next adjacent move to non Y-memory.
	
	Here are examples of the two cases (where x:(r1) is a peripheral):
	
	Example 1:
	
	`move #$12,y0`
	`move x0,x:(r7) y0,y:(r3) (while x:(r7) is a peripheral).`
	
	Example 2:
	
	`mac x1,y0,a x1,x:(r1)+ y:(r6)+,y0`
	`move y0,y1`
	
	This is not a bug, but a documentation update. Any of the following alternatives can be used:
	
	a. Separate these two consecutive moves by any other instruction.
	
	b. Split XY Data Move to two moves.
		DSP563xx Documentation Update

ED3	Description (added 10/09/1997): A one-word conditional branch instruction at LA-1 is not allowed. This is not a bug, but a documentation update.	**DSP563xx Documentation Update**
ED4	Description (added 10/09/1997): The following instructions should not start at address LA: MOVE to/from Program space {MOVEM, MOVEP (only the P space options)} This is not a bug but a documentation update (Appendix B, DSP56300 Family Manual).	**DSP563xx Documentation Update**

Errata No.	Description	Status
ED7	Description (added 1/27/98): When activity passes from one DMA channel to another and the DMA interface accesses external memory (which requires one or more wait states), the DACT and DCH status bits in the DMA Status Register (DSTR) may indicate improper activity status for DMA Channel 0 (DACT = 1 and DCH[2:0] = 000). Workaround: None. **Pertains to:** DSP56300 Family Manual, Sections 8.1.6.3 and 8.1.6.4	**DSP563xx Documentation Update**
ED9	Description (added 1/27/98): When the SCI is configured in Synchronous mode, internal clock, and all the SCI pins are enabled simultaneously, an extra pulse of 1 DSP clock length is provided on the SCLK pin. Workaround: a. Enable an SCI pin other than SCLK. b. In the next instruction, enable the remaining SCI pins, including the SCLK pin. **Pertains to:** UM, SCI Chapter (Use the 302 UM as your reference, Section 8.4.2, “SCI Initialization”)	**DSP563xx Documentation Update**

Errata No.	Description	Status
ED14	The data sheets of the various DSP56300 host interfaces (HI32 excluded) must be modified to make the HI08 / HDI08 compatible with PortA timing 114, which is included here as a reference. Timing 321 "Write data strobe deassertion width" should be split (similar to timing 319 "Read data strobe deassertion width"), as described here: Write data strobe deassertion width: • after HCTR, HCVR and "Last Data Register" writes 2.5*Tc+10.0 @66MHz 2.5*Tc+8.3 @80MHz 2.5*Tc+6.6 @100MHz • after TXH:TXM writes (with HBE=0), TXM:TXL writes (with HBE=1)25 @66MHz 20.6 @80MHz 16.5 @100MHz That is, a minimum of 4 WS for PortA is required for 100 MHz operation. Reference: Timing 114 @ 100MHz 114 WR_ deassertion time 0.5 x TC - 4.0 1.5ns [WS = 1] TC - 2.0 6ns [2 ≤ WS ≤ 3] 2.5 x TC - 4.0 21ns [4 ≤ WS ≤ 7] 3.5 x TC - 4.0 31ns [WS Š 8] **Pertains to**: Data Sheets; timing 114 is in the table on SRAM Read and Write Accesses in the Port A section entitled “External Memory Expansion Port (Port A). The table number is 2-8. Timings 321 and 319 are in the section on Host Interface Timing, Table 2-20.	**DSP563xx Documentation Update**

Errata No.	Description	Status
ED15	Description (added 7/21/98): The DRAM Control Register (DCR) should not be changed while refresh is enabled. If refresh is enabled only a write operation that disables refresh is allowed. Workaround: First disable refresh by clearing the BREN bit, than change other bits in the DCR register, and finally enable refresh by setting the BREN bit.	**DSP563xx Documentation Update**
ED17	Description (added 9/28/98): In all DSP563xx technical datasheets, a note is to be added under "AC Electrical Characteristics" that although the minimum value for "Frequency of Extal" is 0MHz, the device AC test conditions are 15MHz and rated speed. Workaround: N/A	**DSP563xx Documentation Update**
ED20	Description (added 11/24/98): In the Technical Datasheet Voh-TTL should be listed at 2.4 Volts, not as: `TTL = Vcc-0.4` Workaround: This is a documentation update.	**DSP563xx Documentation Update**
ED21	Description (added 11/24/98): In the Technical Datasheet Iol should be listed as 1.6 mA, not as 3.0 mA. Workaround: This is a documentation update.	**DSP563xx Documentation Update**
ED24	Description (added 11/24/98): The technical datasheet supplies a maximum value for internal supply current in Normal, Wait, and Stop modes. These values will be removed because we will specify only a "Typical" current. Workaround: This is a documentation update.	**DSP563xx Documentation Update**

Errata No.	Description	Status
ED26	Description (added 1/6/99): The specification DMA Chapter is wrong. “Due to the DSP56300 Core pipeline, after DE bit in DCRx is set, the corresponding DTDx bit in DSTR will be cleared only after two instruction cycles.” Should be replaced with: “Due to the DSP56300 Core pipeline, after DE bit in DCRx is set, the corresponding DTDx bit in DSTR will be cleared only after three instruction cycles.”	**DSP563xx Documentation Update**

ED28	Description (added 1/7/1997; identified as Documentation Errata 2/1/99):
	
	When two consecutive LAs have a conditional branch instruction at LA-1 of the internal loop, the part does not operate properly. For example, the following sequence may generate incorrect results:
	
	`DO #5, LABEL1`
	`NOP`
	`DO #4, LABEL2`
	`NOP`
	`MOVE (R0) +`
	`BSCC _DEST ; conditional branch at LA-1 of internal loop`
	`NOP ; internal LA LABEL2`
	`NOP ; external LA LABEL1`
	`NOP`
	`NOP`
	`_DEST NOP`
	`NOP`
	`RTS`
	
	Workaround: Put an additional NOP between LABEL2 and LABEL1.
	
	Pertains to: DSP56300 Family Manual, Appendix B, Section B-4.1.3, “At LA-1.”
		DSP563xx Documentation Update

Errata No.	Description	Status
ED29	Description (added 9/12/1997; identified as a Documentation errata 2/1/99): When the ESSI transmits data with the CRA Word Length Control bits (WL[2:0]) = 100, the ESSI is designed to duplicate the last bit of the 24-bit transmission eight times to fill the 32-bit shifter. Instead, after shifting the 24-bit word correctly, eight 0s are being shifted. Workaround: None at this time. **Pertains to:** UM, Section 7.4.1.7, "CRA Word Length Control." The table number is 7-2.	**DSP563xx Documentation Update**
ED30	Description (added 9/12/1997; identified as a Documentation errata 2/1/99): When the ESSI transmits data in the On-Demand mode (that is, MOD = 1 in CRB and DC[4:0] = $00000 in CRA) with WL[2:0] = 100, the transmission does not work properly. Workaround: To ensure correct operation, do not use the On-Demand mode with the WL[2:0] = 100 32-bit Word-Length mode. **Pertains to:** UM, Section 7.5.4.1, "Normal/On-Demand Mode Selection."	**DSP563xx Documentation Update**
ED31/ ES40	Description: Programming the ESSI to use an internal frame sync (that is, SCD2 = 1 in CRB) causes the SC2 and SC1 signals to be programmed as outputs. If however, the corresponding multiplexed pins are programmed by the Port Control Register (PCR) to be GPIOs, then the GPIO Port Direction Register (PRR) chooses their direction, but this causes the ESSI to use an external frame sync if GPI is selected. **Note:** This errata and workaround apply to both ESSI0 and ESSI1. Workaround: To assure correct operation, either program the GPIO pins as outputs or configure the pins in the PCR as ESSI signals. **Note:** The default selection for these signals after reset is GPI.	**DSP563xx Documentation Update**

Errata No.	Description	Status

ED32/ ES103	Description (added 11/9/98):
	
	When returning from a long interrupt (by RTI instruction), and the first instruction after the RTI is a move to a DALU register (A, B, X, Y), the move may not be correct, if the 16-bit arithmetic mode bit (bit 17 of SR) is changed due to the restoring of SR after RTI.
	
	Workaround:
	
	Replace the RTI with the following sequence:
	
	`movec  ssl,sr`
	`nop`
	`rti`
		DSP563xx Documentation Update

Errata No.	Description	Status

ED33/ ES109	Description (added 12/16/98):
	
	When Stack Extension mode is enabled, a use of the instructions BRKcc or ENDDO inside do loops might cause an improper operation.
	
	If the loop is non nested and has no nested loop inside it, the erratais relevant only if LA or LC values are being used outside the loop.
	
	Workaround:
	
	If Stack Extension is used, emulate the BRKcc or ENDDO as in the following examples. We split between two cases, finite loops and do forever loops.
	
	1) Finite DO loops (that is, not DO FOREVER loops)
	
	==========
	
	BRKcc
	
	Original code:
	
	`do #N,label1`
	`.....`
	`.....`
	`do #M,label2`
	`.....`
	`.....`
	`BRKcc`
	`.....`
	`.....`
	`label2`
	`.....`
	`.....`
	`label1`
	
	Will be replaced by:
	
	`do #N, label1`
	`.....`
	`.....`
	`do #M, label2`
	`.....`
	`.....`
	`Jcc fix_brk_routine`
	`.....`
	`.....`
		DSP563xx Documentation Update

Errata No.	Description	Status
ED33/ ES109 cont.	nop_before_label2 nop ; This instruction must be NOP. label2 .... label1 .... fix_brk_routine move #1,lc jmp nop_before_label2 ENDDO ------ Original code: do #M,label1 .... do #N,label2 .... ENDDO .... label2 .... label1 Will be replaced by: do #M, label1 .... do #N, label2 .... JMP fix_enddo_routine	**DSP563xx Documentation Update**

Errata No.	Description	Status
ED33/ ES109 cont.	nop_after_jmp NOP ; This instruction must be NOP. .... label2 .... label1 .... fix_enddo_routine move #1,lc move #nop_after_jmp,la jmp nop_after_jmp 2) DO FOREVER loops ========== BRKcc ----- Original code: do #M,label1 .... do forever,label2 .... BRKcc .... label2 .... label1	**DSP563xx Documentaiton Update**

Errata No.	Description	Status

ED33/ ES109 cont.	Will be replaced by:
	
	`do #M,label1`
	`.....`
	`.....`
	`do forever,label2`
	`.....`
	`.....`
	`JScc fix_brk_forever_routine ; <---`
	`note: JScc and not Jcc`
	`.....`
	`.....`
	
	`nop_before_label2`
	`nop ; This instruction must be NOP.`
	`label2`
	`.....`
	`.....`
	`label1`
	`....`
	`....`
	
	`fix_brk_forever_routine`
	`move ssh,x:<..> ; <..> is some reserved not`
	`used`
	`address (for temporary data)`
	`move #nop_before_label2,ssh`
	`bclr #16,ssl ;`
	`move #1,lc`
	`rti ; <---- note: "rti" and not`
	`"rts" !`
	
	`ENDDO`
	`------`
	`Original code:`
	
	`do #M,label1`
	`.....`
	`.....`
		DSP563xx Documentation Update

Errata No.	Description	Status
ED33/ ES109 cont.	do forever,label2 .... ENDDO .... label2 .... label1 Will be replaced by: do #M,label1 .... do forever,label2 .... JSR fix_enddo_routine ; <--- note: JSR and not JMP nop_after_jmp NOP ; This instruction should be NOP .... label2 .... label1 .... fix_enddo_routine nop move #1,lc bclr #16,ssl move #nop_after_jmp,la rti ; <--- note: "rti" and not "rts"	**DSP563xx Documentation Update**

Errata No.	Description	Status
ED34/ ES110	Description (added 1/5/99): When stack extansion is enabled, the read result from stack may be improper if two previous executed instructions cause sequential read and write operations with SSH. Two cases are possible: Case 1: For the first executed instruction: move from SSH or bit manipulation on SSH (that is, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr). For the second executed instruction: move to SSH or bit manipulation on SSH (that is, jsr, bsr, jscc, bscc). For the third executed instruction: an SSL or SSH read from the stack result may be improper - move from SSH or SSL or bit manipulation on SSH or SSL (that is, bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr). Workaround: Add two NOP instructions before the third executed instruction. Case 2: For the first executed instruction: bit manipulation on SSH (that is, bset, bclr, bchg). For the second executed instruction: an SSL or SSH read from the stack result may be improper - move from SSH or SSL or bit manipulation on SSH or SSL (that is, bset, bclr, bchg, jclr, brclr, jset, brset, btst, bsset, jsset, bsclr, jsclr). Workaround: Add two NOP instructions before the second executed instruction.	**DSP563xx Documentation Update**
ED38	Description (added 7/14/99): If Port A is used for external accesses, the BAT bits in the AAR3-0 registers must be initialized to the SRAM access type (that is, BAT = 01) or to the DRAM access type (that is, BAT = 10). To ensure proper operation of Port A, this initialization must occur even for an AAR register that is not used during any Port A access. Note that at reset, the BAT bits are initialized to 00. **Pertains to:** *DSP56300 Family Manual*, Port A Chapter (Chapter 9 in Revision 2), description of the BAT[1 –0] bits in the AAR3 - AAR0 registers. Also pertains to the core chapter in device-specific user's manuals that include a description of the AAR3 - AAR0 registers with bit definitions (usually Chapter 4).	**DSP563xx Documentation Update**

Errata No.	Description	Status

ED40	Description (added 11/11/99):
	When an instruction with all the following conditions follows a repeat instruction, then the last move will be corrupted.:
	1. The repeated instruction is from external memory.
	2. The repeated instruction is a DALU instruction that includes 2 DALU registers, one as a source, and one as destination (e.g. tfr, add).
	3. The repeated instruction has a double move in parallel to the DALU instruction: one move's source is the destination of the DALU instruction (causing a DALU interlock); the other move's destination is the source of the DALU instruction.
	Example:
	rep #number
	tfr x0,a x(r0)+,x0 a,y0 ; This instruction is from external memory
	|__|_____|------|----------> This is condition 3 second part.
	|_____|----------> This is condition 3, first part - DALU interlock
	In this example, the second iteration before the last, the "x(r0)+,x0" doesn't happen. On the first iteration before the last, the X0 register is fixed with the "x(r0)+,x0", but the "tfr x0,a" gets the wrong value from the previous iteration's X0. Thus, at the last iteration the A register is fixed with "tfr x0,a", but the "a,y0" transfers the wrong value from the previous iteration's A register to Y0.
	Workaround:
	1. Use the DO instruction instead; mask any necessary interrupts before the DO.
	2. Run the REP instructions from internal memory.
	3. Don't make DALU interlocks in the repeated instruction. After the repeat make the move. In the example above, all the "move a,y0" are redundant so it can be done in the next instruction:
	`rep #number`
	`tfr x0,a x(r0)+,x0`
	`move a,y0`
	If no interrupts before the move is a must, mask the interrupts before the REP. **Pertains to:** *DSP56300 Family Manual*, Rev. 2, Section A.3, "Instruction Sequence Restrictions."
		DSP563xx Documentation Update

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations not listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GMBH
Technical Information Center
Schatzbogen 7
81829 München, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
+800 2666 8080

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Order No.: EB343
Rev. 3
10/2005